THE UNIVERSITY OF CHICAGO

Section 9

Fundamentals of Algorithms for Constrained Optimization

Follows N & W, section 15.

9.1 TYPES OF CONSTRAINED OPTIMIZATION ALGORITHMS

Types of Optimization Algorithms

- All of the algorithms solve iteratively a simpler problem.
  - Penalty and Augmented Lagrangian Methods.
  - Sequential Quadratic Programming.
  - Interior-point Methods.
- The approach follows the usual divide-and-conquer approach:
    - Constrained Optimization-
    - Unconstrained Optimization
    - Nonlinear Equations
    - Linear Equations

Quadratic Programming Problems

- Algorithms for such problems are interested to explore because
  - 1. Their structure can be efficiently exploited.
  - 2. They form the basis for other algorithms, such as augmented Lagrangian and Sequential quadratic programming problems.

$$
\begin{aligned}
\min_{x} \quad & q(x) = \tfrac{1}{2} x^T G x + x^T c \\
\text{subject to} \quad & a_i^T x = b_i, \qquad i \in \mathcal{E}, \\
& a_i^T x \geq b_i, \qquad i \in \mathcal{I},
\end{aligned}
$$

Penalty Methods

- Idea: Replace the constraints by a penalty term.
- Inexact penalties: parameter driven to infinity to recover solution. Example:

$$x^* = \arg\min f(x) \text{ subject to } c(x) = 0 \Leftrightarrow$$

$$x^\mu = \arg\min \boxed{f(x) + \frac{\mu}{2}\sum_{i\in\mathcal{E}} c_i^2(x)};\ x^* = \lim_{\mu\to\infty} x^\mu = x^*$$

Solve with unconstrained optimization

- Exact but nonsmooth penalty – the penalty parameter can stay finite.

$$x^* = \arg\min f(x) \text{ subject to } c(x) = 0 \Leftrightarrow x^* = \arg\min f(x) + \mu\sum_{i\in\mathcal{E}} \left|c_i(x)\right|;\ \mu \geq \mu_0$$

Augmented Lagrangian Methods

- Mix the Lagrangian point of view with a penalty point of view.

$$x^* = \arg\min f(x) \text{ subject to } \mathrm{c}(x) = 0 \Leftrightarrow$$

$$x^{\mu,\lambda} = \arg\min f(x) - \sum_{i \in \mathcal{E}} \lambda_i c_i(x) + \frac{\mu}{2} \sum_{i \in \mathcal{E}} c_i^2(x) \Rightarrow$$

$$x^* = \lim_{\lambda \to \lambda^*} x^{\mu,\lambda} \text{ for some } \mu \geq \mu_0 > 0$$

Sequential Quadratic Programming Algorithms

- Solve successively Quadratic Programs.

$$
\begin{aligned}
\min_p \quad & \frac{1}{2} p^T B_k p + \nabla f(x_k) \\
\text{subject to} \quad & \nabla c_i(x_k) d + c_i(x_k) = 0 \quad i \in \mathcal{E} \\
& \nabla c_i(x_k) d + c_i(x_k) \geq 0 \quad i \in \mathcal{I}
\end{aligned}
$$

- It is the analogous of Newton's method for the case of constraints if $B_k = \nabla_{xx}^2 \mathcal{L}(x_k, \lambda_k)$
- But how do you solve the subproblem? It is possible with extensions of simplex which I do not cover.
- An option is BFGS which makes it convex.

Interior Point Methods

- Reduce the inequality constraints with a barrier

$$\begin{aligned} \min_{x,s} \quad & f(x) - \mu \sum_{i=1}^{m} \log s_i \\ \text{subject to} \quad & c_i(x) = 0 && i \in \mathcal{E} \\ & c_i(x) - s_i = 0 && i \in \mathcal{I} \end{aligned}$$

- An alternative, is use a penalty as well:

$$\min_x f(x) - \mu \sum_{i \in \mathcal{I}} \log s_i + \frac{1}{2\mu} \sum_{i \in \mathcal{I}} \left(c_i(x) - s\right)^2 + \frac{1}{2\mu} \sum_{i \in \mathcal{E}} \left(c_i(x)\right)^2$$

- And I can solve it as a sequence of unconstrained problems!

9.2 MERIT FUNCTIONS AND FILTERS

Feasible algorithms

- If I can afford to maintain feasibility at all steps, then I just monitor decrease in objective function.
- I accept a point if I have enough descent.
- But this works only for very particular constraints, such as linear constraints or bound constraints (and we will use it).
- Algorithms that do that are called feasible algorithms.

Infeasible algorithms

- But, sometimes it is VERY HARD to enforce feasibility at all steps (e.g. nonlinear equality constraints).
- And I need feasibility only in the limit; so there is benefit to allow algorithms to move on the outside of the feasible set.
- But then, how do I measure progress since I have two, apparently contradictory requirements:
  - Reduce infeasibility (e.g. $\sum_{i\in\mathcal{E}}|c_i(x)|+\sum_{i\in\mathcal{I}}\max\{-c_i(x),0\}$)
  - Reduce objective function.
  - It has a multiobjective optimization nature!

9.2.1 MERIT FUNCTIONS

Merit function

- One idea also from multiobjective optimization: minimize a weighted combination of the 2 criteria.

$$\phi(x) = w_1 f(x) + w_2 \left[\sum_{i \in \mathcal{E}} |c_i(x)| + \sum_{i \in \mathcal{I}} \max\{-c_i(x), 0\} \right]; \quad w_1, w_2 > 0$$

- But I can scale it so that the weight of the objective is 1.
- In that case, the weight of the infeasibility measure is called "penalty parameter".
- I can monitor progress by ensuring that $\phi(x)$ decreases, as in unconstrained optimization.

Nonsmooth Penalty Merit Functions

$$\phi_1(x;\mu) = f(x) + \mu \sum_{i\in\mathcal{E}} |c_i(x)| + \mu \sum_{i\in\mathcal{I}} [c_i(x)]^-, \qquad [z]^- = \max\{0, -z\}.$$

Penalty parameter

- It is called the l1 merit function.
- Sometimes, they can be even EXACT.

Definition 15.1 (Exact Merit Function).

A merit function $\phi(x;\mu)$ is exact if there is a positive scalar μ^* such that for any $\mu > \mu^*$, any local solution of the nonlinear programming problem (15.1) is a local minimizer of $\phi(x;\mu)$.

We show in Theorem 17.3 that, under certain assumptions, the ℓ_1 merit function $\phi_1(x;\mu)$ is exact and that the threshold value μ^* is given by

$$\mu^* = \max\{|\lambda_i^*|,\ i \in \mathcal{E} \cup \mathcal{I}\},$$

Smooth and Exact Penalty Functions

- Excellent convergence properties, but very expensive to compute.
- Fletcher's augmented Lagrangian:

$$\phi_F(x;\mu) = f(x) - \lambda(x)^T c(x) + \frac{1}{2}\mu \sum_{i\in\mathcal{E}} c_i(x)^2,$$

$$\lambda(x) = [A(x)A(x)^T]^{-1} A(x) \nabla f(x).$$

- It is both smooth and exact, but perhaps impractical due to the linear solve.

Augmented Lagrangian

- Smooth, but inexact.

$$\phi(x) = f(x) - \sum_{i \in \mathcal{E}} \lambda_i c_i(x) + \frac{\mu}{2} \sum_{i \in \mathcal{E}} c_i^2(x) \Rightarrow$$

- An update of the Lagrange Multiplier is needed.
- We will not uses it, except with Augmented Lagrangian methods themselves.

Line-search (Armijo) for Nonsmooth Merit Functions

$$\phi_1(x;\mu) = f(x) + \mu \sum_{i\in\mathcal{E}} |c_i(x)| + \mu \sum_{i\in\mathcal{I}} [c_i(x)]^-,$$

- How do we carry out the "progress search"?
- That is the line search or the sufficient reduction in trust region?
- In the unconstrained case, we had

$$f(x_k) - f(x_k + \beta^m d_k) \geq -\rho\beta^m \nabla f(x_k)^T d_k; \quad 0 < \beta < 1, 0 < \rho < 0.5$$

- But we cannot use this anymore, since the function is not differentiable.

Directional Derivatives of Nonsmooth Merit Function

$$\phi_1(x;\mu) = f(x) + \mu \sum_{i\in\mathcal{E}} |c_i(x)| + \mu \sum_{i\in\mathcal{I}} [c_i(x)]^-,$$

- Nevertheless, the function has a directional derivative (follows from properties of max function). EXPAND

$$D(\phi(x,\mu);p) = \lim_{t\to 0, t>0} \frac{\phi(x+tp,\mu)-\phi(x,\mu)}{t}; \quad D(\max\{f_1,f_2\},p) = \max\{\nabla f_1 p, \nabla f_1 p\}$$

- Line Search: $\phi(x_k,\mu)-\phi(x_k+\beta^m p_k,\mu) \geq -\rho\beta^m D(\phi(x_k,\mu),p_k);$
- Trust Region

$$\phi(x_k,\mu)-\phi(x_k+\beta^m p_k,\mu) \geq -\eta_1(m(0)-m(p_k));$$
$$0<\eta_1<0.5$$

And …. How do I choose the penalty parameter?

- VERY tricky issue, highly dependent on the penalty function used.
- For the l1 function, guideline is:

$$\mu^* = \max\{|\lambda_i^*|,\ i \in \mathcal{E} \cup \mathcal{I}\},$$

- But almost always adaptive. Criterion: If optimality gets ahead of feasibility, make penalty parameter more stringent.
- E.g l1 function: the max of current value of multipliers plus safety factor (EXPAND)
  –

9.2.2 FILTER APPROACHES

Principles of filters

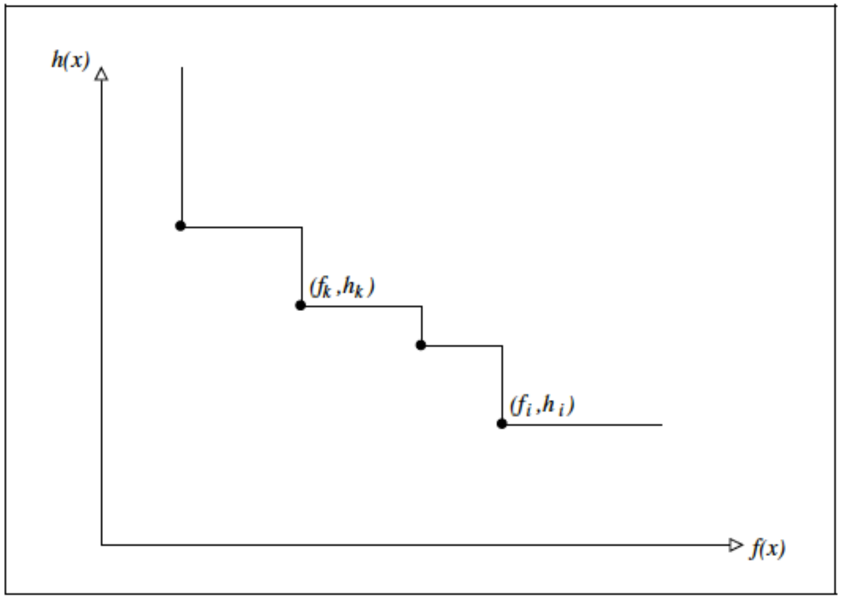
- Originates in the multiobjective optimization philosophy: objective and infeasibility

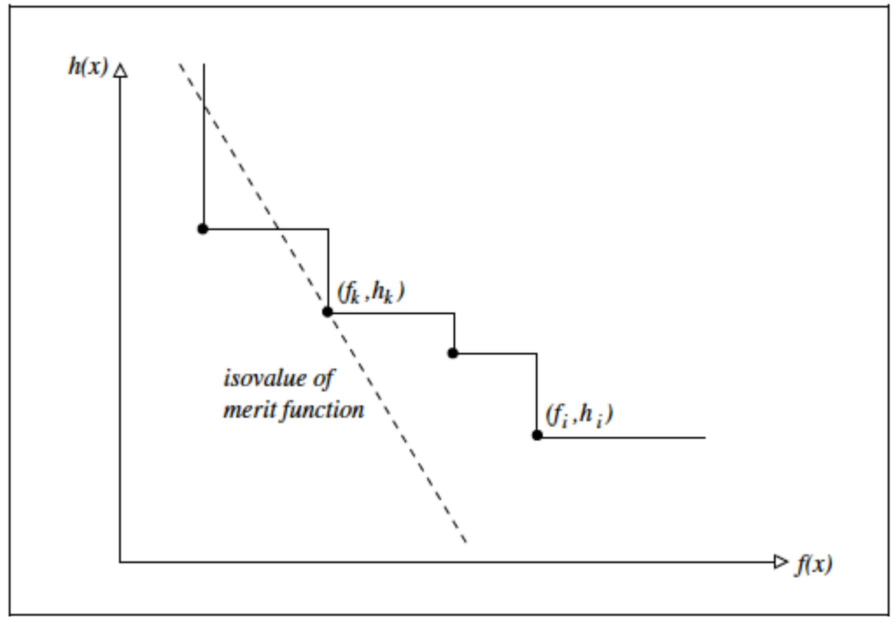
$$h(x) = \sum_{i \in \mathcal{E}} |c_i(x)| + \sum_{i \in \mathcal{I}} [c_i(x)]^-,$$

- The problem becomes:

$$\min_x f(x) \quad \text{and} \quad \min_x h(x).$$

The Filter approach





Definition 15.2.

(a) *A pair* (f_k, h_k) *is said to dominate another pair* (f_l, h_l) *if both* $f_k \le f_l$ *and* $h_k \le h_l$.

(b) *A* filter *is a list of pairs* (f_l, h_l) *such that no pair dominates any other.*

(c) *An iterate* x_k *is said to be acceptable to the filter if* (f_k, h_k) *is not dominated by any pair in the filter.*

Some Refinements

- Like in the line search approach, I cannot accept EVERY decrease since I may never converge.
- Modification:

A trial iterate x^+ is acceptable to the filter if, for all pairs (f_j, h_j) in the filter, we have that

$$f(x^+) \le f_j - \beta h_j \qquad \text{or} \qquad h(x^+) \le h_j - \beta h_j, \qquad \beta \sim 10^{-5} \quad (15.33)$$

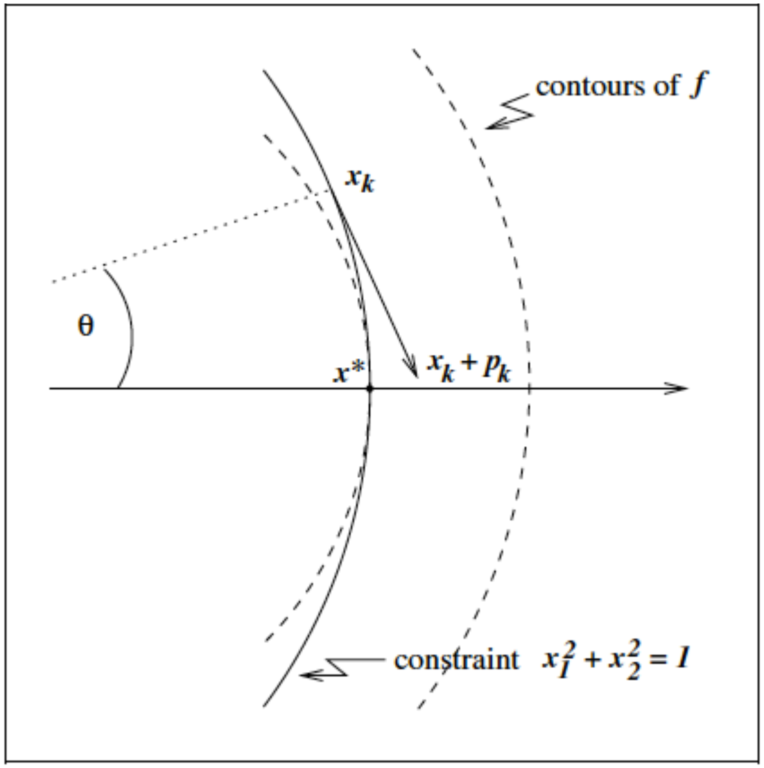
9.3 MARATOS EFFECT AND CURVILINEAR SEARCH

Unfortunately, the Newton step may not be compatible with penalty

- This is called the Maratos effect.
- Problem:

$$\min \ f(x_1, x_2) = 2(x_1^2 + x_2^2 - 1) - x_1,$$

$$x_1^2 + x_2^2 - 1 = 0.$$



- Note: the closest point on search direction (Newton) will be rejected !
- So fast convergence does not occur

Solutions?

- Use Fletcher's function that does not suffer from this problem.
- Following a step: $A_k p_k + c(x_k) = 0.$
- Use a correction that satisfies $A_k \hat{p}_k + c(x_k + p_k) = 0.$

$$\hat{p}_k = -A_k^T (A_k A_k^T)^{-1} c(x_k + p_k),$$

- Followed by the update or line search:

$$x_k + p_k + \hat{p}_k \qquad\qquad x_k + \tau p_k + \tau^2 \hat{p}_k$$

- Since $c(x_k + p_k + \hat{p}_k) = O\left(\|x_k - x^*\|^3\right)$ compared to $c(x_k + p_k) = O\left(\|x_k - x^*\|^2\right)$ corrected Newton step is likelier to be accepted.

THE UNIVERSITY OF CHICAGO

Section 11

Algorithms for Nonlinear Optimization.

Follows N & W, 17 and 19.

Algorithms for constrained optimization

- It is the story of putting ALL these blocks together.
- Augmented Lagrangian
- Interior Point
- Sequential Quadratic Programming

11.1 AUGMENTED LAGRANGIAN

AUGLAG: Equality Constraints

- The augmented Lagrangian:

$$\mathcal{L}_A(x, \lambda; \mu) \stackrel{\text{def}}{=} f(x) - \sum_{i \in \mathcal{E}} \lambda_i c_i(x) + \frac{\mu}{2} \sum_{i \in \mathcal{E}} c_i^2(x),$$

- Observation: if

$$\lambda = \lambda^*; \mu \geq \mu_0 \Rightarrow \nabla_x \mathcal{L}_A\left(x^*, \lambda^*, \mu\right) = 0;$$

$$\nabla_{xx}^2 \mathcal{L}_A\left(x^*, \lambda^*, \mu\right) = \nabla_{xx}^2 \mathcal{L}\left(x^*, \lambda^*, \mu\right) + \mu\left(\nabla c\left(x^*\right)\right)^T \left(\nabla c\left(x^*\right)\right)$$

AUGLAG: SOC

- So x* is a stationary point for Auglag for exact multipliers … but is it a minimum?
- Yes, for mu sufficiently large.

$$\nabla^2_{xx}\mathcal{L}_{\mathcal{A}}\left(x^*,\lambda^*,\mu\right) \sim \begin{bmatrix} Y & Z \end{bmatrix}^T \nabla^2_{xx}\mathcal{L}_{\mathcal{A}}\left(x^*,\lambda^*,\mu\right)\begin{bmatrix} Y & Z \end{bmatrix} + \mu\left(\nabla c\left(x^*\right)Y\right)^T\left(\nabla c\left(x^*\right)Y\right) =$$

$$\begin{bmatrix} Z^T\nabla^2_{xx}\mathcal{L}_{\mathcal{A}}\left(x^*,\lambda^*,\mu\right)Z & * \\ * & * + \mu\left(\nabla c\left(x^*\right)Y\right)^T\left(\nabla c\left(x^*\right)Y\right) \end{bmatrix} \succ 0 \quad \text{for } \mu \text{ suff large.}$$

- So it is *almost* as solving unconstrained problem … but how do I find multiplier estimates?

Multiplier Estimates Auglag

- At the current estimate, solve problem

$$0 \approx \nabla_x \mathcal{L}_A(x_k, \lambda^k; \mu_k) = \nabla f(x_k) - \sum_{i \in \mathcal{E}} [\lambda_i^k - \mu_k c_i(x_k)] \nabla c_i(x_k).$$

- The obvious choice:

$$\lambda_i^{k+1} = \lambda_i^k - \mu_k c_i(x_k), \qquad \text{for all } i \in \mathcal{E}.$$

- What do I do if I converge lambda but x* is not feasible? Increase the penalty mu (it will have to end increasing eventually).

The general case

- The bound constrained formulation. Slacks.

$$c_i(x) \geq 0, i \in \mathcal{I}, \quad \Longrightarrow \quad c_i(x) - s_i = 0, \quad s_i \geq 0, \quad \text{for all } i \in \mathcal{I}.$$

- The problem:

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{subject to} \quad c_i(x) = 0, \; i = 1, 2, \ldots, m, \; l \leq x \leq u.$$

The augmented Lagrangian

- The new AugLag

$$\mathcal{L}_A(x, \lambda; \mu) = f(x) - \sum_{i=1}^{m} \lambda_i c_i(x) + \frac{\mu}{2} \sum_{i=1}^{m} c_i^2(x).$$

- The bound constrained optimization problem:

$$\min_x \mathcal{L}_A(x, \lambda; \mu) \quad \text{subject to } l \leq x \leq u.$$

- Same property: if Lagrange multiplier is the optimal one for eq cons and mu is large enough then x* is a solution !

Practical AugLag alg: LANCELOT

Main computation: Use bound constrained projection.

Forcing sequences

Algorithm 17.4 (Bound-Constrained Lagrangian Method).

Choose an initial point x_0 and initial multipliers λ^0;
Choose convergence tolerances η_* and ω_*;
Set $\mu_0 = 10$, $\omega_0 = 1/\mu_0$, and $\eta_0 = 1/\mu_0^{0.1}$;
for $k = 0, 1, 2, \ldots$
  Find an approximate solution x_k of the subproblem (17.50) such that

$$\left\| x_k - P\left(x_k - \nabla_x \mathcal{L}_A(x_k, \lambda^k; \mu_k), l, u\right) \right\| \le \omega_k;$$

  if $\|c(x_k)\| \le \eta_k$
    (* test for convergence *)
    if $\|c(x_k)\| \le \eta_*$ and $\left\| x_k - P\left(x_k - \nabla_x \mathcal{L}_A(x_k, \lambda^k; \mu_k), l, u\right) \right\| \le \omega_*$
      stop with approximate solution x_k;
    end (if)
    (* update multipliers, tighten tolerances *)
    $\lambda^{k+1} = \lambda^k - \mu_k c(x_k)$;
    $\mu_{k+1} = \mu_k$;
    $\eta_{k+1} = \eta_k / \mu_{k+1}^{0.9}$;
    $\omega_{k+1} = \omega_k / \mu_{k+1}$;
  else
    (* increase penalty parameter, tighten tolerances *)
    $\lambda^{k+1} = \lambda^k$;
    $\mu_{k+1} = 100\mu_k$;
    $\eta_{k+1} = 1/\mu_{k+1}^{0.1}$;
    $\omega_{k+1} = 1/\mu_{k+1}$;
  end (if)
end (for)

Solving the bound constrained subproblem

- It is an iterative bound constrained optimization algorithm with trust-region:

$$\min_d \tfrac{1}{2} d^T \left[\nabla^2_{xx}\mathcal{L}(x_k, \lambda^k) + \mu_k A_k^T A_k\right] d + \nabla_x \mathcal{L}_A(x_k, \lambda^k; \mu_k)^T d$$
$$\text{subject to } \; l \le x_k + d \le u, \qquad \|d\|_\infty \le \Delta,$$

- Each step solves a bound constrained QP (not necessarily PD), same as in your homework 4.
- The difference: after a subspace solve: compute the new derivative and update TR.

11.2 INTERIOR-POINT METHODS

Outline

- Same idea as in the case of the interior-point method for QP.
- Create a path that is interior with respect to the Lagrange multipliers and the slacks that depends on a smoothing parameter mu.
- Drive mu to 0.

Interior –point, “smoothing” parth

- Formulation (with slacks) :

$$
\begin{aligned}
\min_{x,s} \quad & f(x) \\
\text{subject to} \quad & c_E(x) = 0, \\
& c_I(x) - s = 0, \\
& s \geq 0.
\end{aligned}
$$

- Interior-point (smoothing path; mu=0: KKT)

$$
\begin{aligned}
\nabla f(x) - A_E{}^T(x)y - A_I{}^T(x)z &= 0, & c_E(x) &= 0, \\
Sz - \mu e &= 0, & c_I(x) - s &= 0,
\end{aligned}
$$

Barrier interpretation

- The nonlinear equation is the same as the KKT point of the barrier function:

$$
\begin{aligned}
\min_{x,s} \; & f(x) - \mu \sum_{i=1}^{m} \log s_i \\
\text{subject to} \quad & c_{\mathrm{E}}(x) = 0, \\
& c_{\mathrm{I}}(x) - s = 0,
\end{aligned}
$$

Newton Method:

- Linearization for fixed mu:

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L} & 0 & -A_{\mathcal{E}}^T(x) & -A_{\mathcal{I}}^T(x) \\ 0 & Z & 0 & S \\ A_{\mathcal{E}}(x) & 0 & 0 & 0 \\ A_{\mathcal{I}}(x) & -I & 0 & 0 \end{bmatrix} \begin{bmatrix} p_x \\ p_s \\ p_y \\ p_z \end{bmatrix} = - \begin{bmatrix} \nabla f(x) - A_{\mathcal{E}}^T(x)y - A_{\mathcal{I}}^T(x)z \\ Sz - \mu e \\ c_{\mathcal{E}}(x) \\ c_{\mathcal{I}}(x) - s \end{bmatrix},$$

$$\mathcal{L}(x, s, y, z) = f(x) - y^T c_{\mathcal{E}}(x) - z^T(c_{\mathcal{I}}(x) - s).$$

Choose the step

- The new iteration:

$$x^+ = x + \alpha_s^{\max} p_x, \quad s^+ = s + \alpha_s^{\max} p_s,$$
$$y^+ = y + \alpha_z^{\max} p_y, \quad z^+ = z + \alpha_z^{\max} p_z,$$

- Where:

$$\alpha_s^{\max} = \max\{\alpha \in (0, 1] : s + \alpha p_s \geq (1 - \tau)s\},$$
$$\alpha_z^{\max} = \max\{\alpha \in (0, 1] : z + \alpha p_z \geq (1 - \tau)z\},$$

- And,

$$\tau = 0.99 - 0.995$$

How do I measure progress?

- Merit function:

$$E(x, s, y, z; \mu) = \max\left\{\|\nabla f(x) - A_{\mathrm{E}}(x)^T y - A_{\mathrm{I}}(x)^T z\|, \|Sz - \mu e\|, \|c_{\mathrm{E}}(x)\|, \|c_{\mathrm{I}}(x) - s\|\right\},$$

- I try to decrease it as much as I can.

Basic Interior-Point Algorithm

Algorithm 19.1 (Basic Interior-Point Algorithm).

Choose x_0 and $s_0 > 0$, and compute initial values for the multipliers y_0 and $z_0 > 0$. Select an initial barrier parameter $\mu_0 > 0$ and parameters $\sigma, \tau \in (0, 1)$. Set $k \leftarrow 0$.

repeat until a stopping test for the nonlinear program (19.1) is satisfied
    repeat until $E(x_k, s_k, y_k, z_k; \mu_k) \le \mu_k$
        Solve (19.6) to obtain the search direction $p = (p_x, p_s, p_y, p_z)$;
        Compute $\alpha_s^{\max}, \alpha_z^{\max}$ using (19.9);
        Compute $(x_{k+1}, s_{k+1}, y_{k+1}, z_{k+1})$ using (19.8);
        Set $\mu_{k+1} \leftarrow \mu_k$ and $k \leftarrow k + 1$;
    end
    Choose $\mu_k \in (0, \sigma\mu_k)$;
end

How to solve the linear system

- Rewriting the Newton Direction:

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L} & 0 & A_E{}^T(x) & A_I{}^T(x) \\ 0 & \Sigma & 0 & -I \\ A_E(x) & 0 & 0 & 0 \\ A_I(x) & -I & 0 & 0 \end{bmatrix} \begin{bmatrix} p_x \\ p_s \\ -p_y \\ -p_z \end{bmatrix} = - \begin{bmatrix} \nabla f(x) - A_E{}^T(x)y - A_I{}^T(x)z \\ z - \mu S^{-1}e \\ c_E(x) \\ c_I(x) - s \end{bmatrix}$$

$$\Sigma = S^{-1}Z.$$

- Can use indefinite factorization LDLT.
- Or, projected CG (since it is in saddle-point form)

Linear System, part II

- Or, we can eliminate p_s and use LDLT

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L} & A_E^T(x) & A_I^T(x) \\ A_E(x) & 0 & 0 \\ A_I(x) & 0 & -\Sigma^{-1} \end{bmatrix} \begin{bmatrix} p_x \\ -p_y \\ -p_z \end{bmatrix} = -\begin{bmatrix} \nabla f(x) - A_E^T(x)y - A_I^T(x)z \\ c_E(x) \\ c_I(x) - \mu Z^{-1}e \end{bmatrix}$$

- And even p_z:

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L} + A_I^T \Sigma A_I & A_E^T(x) \\ A_E(x) & 0 \end{bmatrix}$$

How do we deal with nonconvexity and non-LICQ?

- Regularization

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L} + \delta I & 0 & A_E(x)^T & A_I(x)^T \\ 0 & \Sigma & 0 & -I \\ A_E(x) & 0 & -\gamma I & 0 \\ A_I(x) & -I & 0 & 0 \end{bmatrix}.$$

- Choose delta so that signature of the matrix corresponds to positive definiteness of reduced matrix: $(n+m, l+m, 0)$
- For signature, can use LDLT

But, how do I know how far to go in a direction?

- Backtracking search for merit function (based on barrier interpretation) :

$$\phi_\nu(x,s) = f(x) - \mu \sum_{i=1}^{m} \log s_i + \nu \|c_{\mathrm{E}}(x)\| + \nu \|c_{\mathrm{I}}(x) - s\|,$$

$$\alpha_s \in (0, \alpha_s^{\max}], \qquad \alpha_z \in (0, \alpha_z^{\max}],$$

- Directional derivative (for line search)

$$\frac{\partial}{\partial p}\|c(x)\| = \frac{\partial}{\partial p}\sqrt{c(x)^T c(x)} = \begin{cases} \dfrac{c(x)}{\|c(x)\|}\nabla c(x)\, p & c(x) \neq 0 \\ \dfrac{\nabla c(x)\, p}{\|\nabla c(x)\, p\|}\nabla c(x)\, p & c(x) = 0, \nabla c(x)\, p \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

How do we update barrier parameter?

- Decrease of barrier (example):

$$\mu_{k+1} = \sigma_k \mu_k, \quad \text{with} \quad \sigma_k \in (0, 1).$$

$$\sigma_k = 0.1 \min\left(0.05\frac{1-\xi_k}{\xi_k}, 2\right)^3, \quad \text{where} \quad \xi_k = \frac{\min_i [s_k]_i [z_k]_i}{(s^k)^T z^k / m}.$$

- Step update:

$$x^+ = x + \alpha_s p_x, \quad s^+ = s + \alpha_s p_s,$$
$$y^+ = y + \alpha_z p_y, \quad z^+ = z + \alpha_z p_z.$$

A practical interior-point algorithm

Algorithm 19.2 (Line Search Interior-Point Algorithm).

Choose x_0 and $s_0 > 0$, and compute initial values for the multipliers y_0 and $z_0 > 0$. If a quasi-Newton approach is used, choose an $n \times n$ symmetric and positive definite initial matrix B_0. Select an initial barrier parameter $\mu > 0$, parameters $\eta, \sigma \in (0, 1)$, and tolerances ϵ_μ and ϵ_{TOL}. Set $k \leftarrow 0$.

repeat until $E(x_k, s_k, y_k, z_k; 0) \le \epsilon_{\text{TOL}}$
    repeat until $E(x_k, s_k, y_k, z_k; \mu) \le \epsilon_\mu$
        Compute the primal-dual direction $p = (p_x, p_s, p_y, p_z)$ from
            (19.12), where the coefficient matrix is modified as in
            (19.25), if necessary;
        Compute $\alpha_s^{\max}, \alpha_z^{\max}$ using (19.9); Set $p_w = (p_x, p_s)$;
        Compute step lengths α_s, α_z satisfying both (19.27) and
         $\phi_\nu(x_k + \alpha_s p_x, s_k + \alpha_s p_s) \le \phi_\nu(x_k, s_k) + \eta \alpha_s D\phi_\nu(x_k, s_k; p_w)$;
        Compute $(x_{k+1}, s_{k+1}, y_{k+1}, z_{k+1})$ using (19.28);
        if a quasi-Newton approach is used
            update the approximation B_k;
        Set $k \leftarrow k + 1$;
    end
    Set $\mu \leftarrow \sigma \mu$ and update ϵ_μ;
end

11.3 SEQUENTIAL QUADRATIC PROGRAMMING

Idea:

- Start with equality constrained problem:

$$\begin{aligned} &\min f(x) \\ &\text{subject to } c(x) = 0, \end{aligned}$$

- Find the solution p_k, l_k of problem with quadratic objective and linearized constraints called quadratic program.

$$\begin{aligned} \min_{p} \quad & f_k + \nabla f_k^T p + \tfrac{1}{2} p^T \nabla_{xx}^2 \mathcal{L}_k p \\ \text{subject to} \quad & A_k p + c_k = 0. \end{aligned}$$

- Define: $\lambda_{k+1} = l_k; x_{k+1} = x_k + p_k$ which gives Newton's.

Extension to inequality constraints.

- For problem: $$\min f(x) \qquad \text{subject to } c_i(x) = 0, \quad i \in \mathcal{E}, \qquad c_i(x) \geq 0, \quad i \in \mathcal{I}.$$

- Solve successively the quadratic program:

$$\begin{aligned} \min_p \quad & f_k + \nabla f_k^T p + \tfrac{1}{2} p^T \nabla_{xx}^2 \mathcal{L}_k p \\ \text{subject to} \quad & \nabla c_i(x_k)^T p + c_i(x_k) = 0, \quad i \in \mathcal{E}, \\ & \nabla c_i(x_k)^T p + c_i(x_k) \geq 0, \quad i \in \mathcal{I}. \end{aligned}$$

- E.g use a BFGS approximation (though density an issue) and interior point (defined in section 10).

A sequential Linear-Quadratic Program

- Analogous with the projection/subspace minimization algorithm.
- In Linear phase, solve (e.g by interior point)

$$\begin{aligned} \min_{p} \quad & f_k + \nabla f_k^T p \quad \left(+ \frac{1}{2} p^T p \right) \\ \text{subject to} \quad & c_i(x_k) + \nabla c_i(x_k)^T p = 0, \quad i \in \mathcal{E}, \\ & c_i(x_k) + \nabla c_i(x_k)^T p \geq 0, \quad i \in \mathcal{I}, \\ & \|p\|_\infty \leq \Delta_k^{\mathrm{LP}}, \end{aligned}$$

- Variation: (infeas)

$$\begin{aligned} \min_{p} \quad & l_\mu(p) \stackrel{\text{def}}{=} f_k + \nabla f_k^T p + \mu \sum_{i \in \mathcal{E}} |c_i(x_k) + \nabla c_i(x_k)^T p| \\ & \qquad + \mu \sum_{i \in \mathcal{I}} [c_i(x_k) + \nabla c_i(x_k)^T p]^- \\ \text{subject to} \quad & \|p\|_\infty \leq \Delta_k^{\mathrm{LP}}. \end{aligned}$$

Determine active set in linear phase

- For feasible algorithm:

$$\mathcal{A}_k(p^{\text{LP}}) = \{i \in \mathcal{E} \mid c_i(x_k) + \nabla c_i(x_k)^T p^{\text{LP}} = 0\} \cup \{i \in \mathcal{I} \mid c_i(x_k) + \nabla c_i(x_k)^T p^{\text{LP}} = 0\}.$$

p^c

- For infeasible algorithm:

$$\mathcal{V}_k(p^{\text{LP}}) = \{i \in \mathcal{E} \mid c_i(x_k) + \nabla c_i(x_k)^T p^{\text{LP}} \neq 0\} \cup \{i \in \mathcal{I} \mid c_i(x_k) + \nabla c_i(x_k)^T p^{\text{LP}} < 0\}.$$

- Backtrack merit function on p^{LP} to obtain Cauchy pt p^C

$$q_\mu(p) \stackrel{\text{def}}{=} f_k + \nabla f_k^T p + \frac{1}{2} p^T \nabla^2_{xx} \mathcal{L}_k p + \mu \sum_{i \in \mathcal{E}} |c_i(x_k) + \nabla c_i(x_k)^T p| + \mu \sum_{i \in \mathcal{I}} [c_i(x_k) + \nabla c_i(x_k)^T p]^-$$

Equality Constrained QP: EQP

- Determine the working active set:

$$\mathcal{W}_k \subset \mathcal{A}_k \left(\text{or } \mathcal{V}_k\right)$$

- Solve EQP:

$$\begin{aligned}
\min_p \quad & f_k + \tfrac{1}{2} p^T \nabla_{xx}^2 \mathcal{L}_k p + \left(\nabla f_k + \mu_k \sum_{i \in \mathcal{V}_k} \gamma_i \nabla c_i(x_k) \right)^T p \\
\text{subject to} \quad & c_i(x_k) + \nabla c_i(x_k)^T p = 0, \quad i \in \mathcal{E} \cap \mathcal{W}_k, \\
& c_i(x_k) + \nabla c_i(x_k)^T p = 0, \quad i \in \mathcal{I} \cap \mathcal{W}_k, \\
& \|p\|_2 \le \Delta_k,
\end{aligned}$$

- E.g by truncated, projected CG.

Total Step

- Start from Cauchy Direction:

$$p_k = p^{C} + \alpha^{Q}(p^{Q} - p^{C}),$$

- Choose α^Q by backtracking using the same merit function as in first stage. (effectively, a dogleg).
- If the LP solution is infeasible, increase the penalty.